

Romain Jacquet

BTS SIO 1

01/10/2025

Compte rendu TP Atelier 05 : TD3 CSS :
HTML/CSS Avancé (2/2)

Sommaire

Introduction.....	2
Exercice 1.....	2
Exercice 2.....	2
Exercice 3.....	3
Exercice 4.....	4
Exercice 5.....	5
Exercice 6.....	6
Exercice 7.....	7
Exercice 8.....	8
Exercice 9.....	9
Exercice 10.....	10
Exercice 11.....	10
Conclusion.....	10

Introduction

Ce TP m'a permis de découvrir concrètement comment fonctionne CSS et comment gérer la priorité des styles. J'ai surtout travaillé sur la création et l'organisation de menus, ainsi que sur l'alignement des éléments avec Flexbox pour rendre la page plus claire et fonctionnelle.

Exercice 1

```
Exercice 1

1. #id
2. nav.titi .tata div div div div div
3. ul li div.skill
4. .titi span
5. div span
6. div > a
7. div + a

(Les sélecteurs 5, 6 et 7 ont la même priorité)
```

Voici le screen(de mon fichier txt notepad)des priorité des sélecteurs

Exercice 2

Cas 1 :

La couleur du texte "Priorité CSS" sera rouge. En effet, lorsque plusieurs règles CSS s'appliquent au même élément, le style inline écrit directement dans la balise HTML est toujours prioritaire. Même si les fichiers CSS externes et internes définissent d'autres couleurs, c'est la déclaration inline qui l'emporte.

```

<html>
<head>
  <style type="text/css">
    p {color: pink}
  </style>
  <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body><p style="color:red">Priorité CSS</p></body>
</html>

```

Cas 2 :

La couleur du texte "Priorité CSS" sera bleue. Dans ce cas, il n'y a pas de style inline, et les deux règles CSS (interne et externe) ont la même spécificité. C'est donc l'ordre de chargement qui décide : la dernière règle lue par le navigateur est appliquée. Comme le fichier CSS externe est chargé après le style interne, la couleur bleue écrase la couleur rose.

```

<html>
<head>
  <style type="text/css">
    p {color: pink}
  </style>
  <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body><p>Priorité CSS</p></body>
</html>

```

Exercice 3

1)

J'ai remplacé l'ancien menu par :

```

<nav>
  <div><a href="./index.html">Accueil</a></div>
  <div><a href="./facts.html">Facts</a></div>
  <div><a href="./news.html">Actualités</a></div>
  <div><a href="./contact.html">Contact</a></div>
</nav>

```

2)

J'ai vérifié que la balise `<nav>` prend bien **toute la largeur de son parent** (car elle est en `display: block` par défaut).

3)

J'ai donné une largeur de **75%** à `<nav>` avec `margin: 0 auto` pour la **centrer horizontalement** :

```
nav {  
  width: 75%;  
  margin: 0 auto;  
}
```

4)

J'ai mis une couleur de fond aux `<div>` enfants directs de `<nav>`

```
nav > div {  
  background-color: #5BBDBF;  
}
```

5) J'ai donné une couleur différente aux liens `<a>`

```
nav a {  
  background-color: #c0d5c2;  
}
```

Exercice 4

1)

J'ai changé les `<div>` dans le menu pour qu'ils soient en `display: inline` :

```
nav > div {  
  display: inline;  
}
```

2)

J'ai supprimé les espaces entre les `<div>` en mettant des commentaires HTML :

```
<div><a href="./index.html">Accueil</a></div><!--  
--><div><a href="./facts.html">Facts</a></div><!-- etc. -->
```

3)

J'ai donné une hauteur de 50px au `<nav>` :

```
nav {  
  height: 50px;  
}
```

4)

J'ai ajouté un padding horizontal de 10px aux liens `<a>` :

```
nav a {  
  padding: 0 10px;  
}
```

5)

J'ai mis une bordure noire à gauche sur chaque lien `<a>` :

```
nav a {  
  border-left: 2px solid #000;  
}
```

6)

J'aurais dû enlever cette bordure pour le premier lien avec :

```
nav a:first-child {  
  border-left: none;  
}
```

Exercice 5

1)

J'ai ajouté des sous-menus après les liens "Accueil" et "Contact", dans le même `<div>` :

```

<div>
  <a href="./index.html">Accueil</a>
  <div class="submenu">
    <div><a href="./one.html">one</a></div>
    <div><a href="./two.html">two</a></div>
    <div><a href="./three.html">three</a></div>
  </div>
</div>

```

2)

J'ai donné la classe `.submenu` à ces sous-menus, et je leur ai appliqué :

```

.submenu {
  position: absolute;
  top: 50px;
  left: 0px;
}

```

Exercice 6

1)

J'ai créé les sous-menus cachés

- J'ai ajouté les sous-menus dans le HTML avec la classe `submenu`
- Dans le CSS, je leur ai donné la couleur de fond `#aca`
- Je les ai masqués avec `display: none`

2)

J'ai activé l'affichage au survol

- J'ai utilisé le sélecteur `nav > div:hover .submenu`
- J'ai mis `display: block` pour les faire apparaître
- Quand je survole "Accueil" ou "Contact", leur sous-menu respectif s'affiche

Exercice 7

1)

J'ai ajouté `display: flex` à l'élément `<nav>` pour transformer le conteneur de navigation en un conteneur flex. Cela permet de mieux contrôler l'alignement et la disposition des éléments de menu.

2)

J'ai changé la valeur du display des `<div>` directement enfants de `<nav>` en `display: inline`. Cette modification a pour effet d'aligner horizontalement les éléments de menu au lieu de les empiler verticalement comme le ferait `display: block`.

3)

J'ai testé `flex-direction: column` sur l'élément `<nav>`, ce qui a eu pour effet de réorganiser les éléments de menu en colonne verticale. Cette disposition n'étant pas souhaitable pour une navigation horizontale, je suis revenu à la valeur par défaut.

4)

J'ai rétabli `flex-direction: row` pour conserver l'alignement horizontal des éléments de menu, qui est la disposition standard pour une navigation principale.

5)

J'ai appliqué une largeur fixe de 100px à tous les `<div>` descendants de `<nav>` en utilisant `width: 100px` pour les menus principaux et `min-width: 100px` pour les sous-menus. Cela garantit que tous les menus et sous-menus ont la même largeur, créant un design plus cohérent et professionnel.

Exercice 8

1)

J'ai commencé par préparer le terrain

J'ai donné à la barre de navigation une hauteur fixe de 200px avec une couleur rose fluo (#FF00FF). J'ai aussi décalé les sous-menus de 200px vers le bas pour mieux voir ce qu'ils faisaient.

C'était un peu moche, mais ça m'a bien aidé à comprendre ce qui n'allait pas.

2) J'ai fait un premier test avec un centrage simple

J'ai mis `align-items: center` sur la nav.

J'ai vu que :

- Les titres comme « Accueil » et « Contact » étaient bien centrés.
- Par contre, les menus ne prenaient pas toute la hauteur.
- Et surtout, les sous-menus étaient trop loin, il fallait passer la souris dans un vide pour les atteindre.

En gros, c'était pas pratique du tout.

3) J'ai fait un deuxième test en étirant

J'ai essayé `align-items: stretch`.

J'ai vu que :

- Les menus prenaient toute la hauteur.
- Les sous-menus étaient plus accessibles.

- Mais le texte restait collé en haut, pas centré.

C'était mieux pour l'accès, mais pas joli à voir.

4) J'ai fait une troisième solution en combinant les deux

J'ai gardé `align-items: stretch` sur la barre principale, puis j'ai transformé chaque menu en conteneur flex et mis `align-items: center` dedans.

Et là, j'ai eu exactement ce que je voulais :

- Les menus prennent toute la hauteur.
- Le texte est bien centré.
- Les sous-menus sont faciles à atteindre.

5) J'ai fait le nettoyage final

J'ai enlevé les couleurs et hauteurs fixes, et tout fonctionnait toujours bien. La barre de navigation était redevenue normale, mais bien mieux structurée.

Exercice 9

Pour cet exercice, je voulais que les titres du menu (comme "Accueil", "Contact", etc.) soient alignés à droite de la barre de navigation.

Pour y arriver, j'ai utilisé Flexbox sur la balise `<nav>` avec la propriété `display: flex;`, puis j'ai ajouté `justify-content: flex-end;` pour pousser tout le menu vers la droite.

J'ai aussi mis une petite couleur de fond temporaire (`#640051`) juste pour bien visualiser le résultat pendant mes tests.

Au final, le menu s'est bien positionné du côté droit, exactement comme je le voulais.

Exercice 10

Ici, le but était de mettre le menu sur la même ligne que la citation qui se trouvait dans le `<header>`.

Pour ça, j'ai simplement transformé le header en conteneur flex avec `display: flex;`.

Grâce à ça, la citation et le menu se sont retrouvés côte à côte sans problème.

Ensuite, j'ai voulu placer le menu en bas du header, alors j'ai ajouté `align-items: flex-end;` pour qu'il soit bien aligné en bas.

Et pour finir, j'ai mis un fond blanc au header (`background-color: white;`) pour que le rendu soit plus propre.

Avec ces trois petits réglages, le menu et la citation étaient bien alignés, sur une seule ligne, avec un style simple et clair.

Exercice 11

Dans ce dernier exercice, j'ai voulu rendre la page un peu plus large et organiser le contenu différemment.

J'ai donc commencé par augmenter la largeur du body, en passant de `600px` à `900px`, histoire que tout respire un peu plus à l'écran.

Ensuite, j'ai appliqué `display: flex;` sur la balise `<main>` pour pouvoir afficher plusieurs blocs côte à côte, comme un article et une barre latérale.

Pour équilibrer la mise en page, j'ai donné `width: 60%` à l'article et `width: 30%` à l'aide, avec une petite marge de 10% pour que ce soit bien espacé.

Enfin, j'ai ajouté une couleur de fond légère à la zone principale, juste pour mieux voir la structure pendant mes tests.

Grâce à ces changements, le site avait un rendu plus clair, plus aéré et surtout plus professionnel.

Conclusion

Grâce à ce TP, j'ai mieux compris comment structurer et styliser une page web. J'ai appris à créer des menus interactifs, à aligner les contenus facilement et à rendre l'ensemble plus lisible et agréable à l'œil.